

Abstract

The ManArray processor is a scalable indirect VLIW array processor that defines two preferred architectures for indirect VLIW memories. One approach treats the VIM as one composite block of memory using one common address interface to access any VLIW stored in the VIM. The second approach treats the VIM as made up of multiple smaller VIMs each individually associated with the functional units and each individually addressable for loading and reading during XV execution. The VIM memories, contained in each processing element (PE), are accessible by the same type of LV and XV Short Instruction Words (SIWs) as in a single processor instantiation of the indirect VLIW architecture. In the ManArray architecture, the control processor, also called a sequence processor (SP), fetches the instructions from the SIW memory and dispatches them to itself and the PEs. By using the LV instruction, VLIWs can be loaded into VIMs in the SP and the PEs. Since the LV instruction is supplied by the SP through the instruction stream, when VLIWs are being loaded into any VIM no other processing takes place. In addition, as defined in the ManArray architecture, when the SP is processing SIWs, such as control and other sequential code, the PE array is not executing any instructions. Techniques are provided herein to independently load the VIMs concurrent with SIW or iVLIW execution on the SP or on the PEs thereby allowing the load latency to be hidden by the computation.